



INSTANT GAME FOR FACEBOOK AND MOBILE

Final Report

Spring 2011

Ásgeir Einarsson

Bergur Páll Gylfason

Daði Magnússon

Vignir Jónsson

B.Sc. in Computer Science

Instructor: Hannes Pétursson
Examiner: Hlynur Sigurpórsson
Partner: Betware

T-404-LOKA
School of Computer Science

TABLE OF CONTENTS

1. Preface	3
2. Introduction	4
3. Artifacts	5
4. Methodology.....	6
5. Project analysis.....	7
5.1. Work Schedule	7
5.2. Project Progress.....	7
5.2.1. Sprint burndown charts explained	7
5.2.2. Planning period (Jan 12 – Jan 31)	7
5.2.3. Sprint 1 (Feb 1 – Feb 14).....	8
5.2.4. Sprint 2 (Feb 15 – Feb 28).....	8
5.2.5. Sprint 3 (Mar 1 – Mar 14)	8
5.2.6. Sprint 4 (Mar 14 – Mar 28).....	9
5.2.7. Sprint 5 (Apr 19 – May 2)	9
5.2.8. Sprint 6 (May 3 – May 16)	9
5.2.9. Release Burndown.....	10
5.3. Risk Analysis.....	11
6. Design and Development	12
6.1. Fisherman	12
6.1.1. Graphics.....	13
6.1.2. Skinnability	14
6.1.3. Facebook	15
6.2. Game Client Framework (GCF).....	16
6.2.1. Breakout	17
6.3. Testing	18
6.3.1. Unit tests	18
6.3.2. User tests.....	18
6.3.3. Acceptance tests.....	18
6.3.4. Results	18
7. Summary	19
8. Conclusion.....	20
9. Bibliography	21

1. PREFACE

This report is a part of a final project in Computer Science from Reykjavik University in the spring semester 2011.

The following students worked on the project:

- Ásgeir Einarsson, SSN: 300586-2959
- Bergur Páll Gylfason, SSN: 070888-3359
- Daði Magnússon, SSN: 051188-2419
- Vignir Jónsson, SSN: 260188-2919

Work on the project started January 10th 2011 and finished with a presentation May 19th 2011, although this document and all other artifacts were handed in May 16th 2011.

This final project is done in collaboration with Betware, with Hlynur Johnsen as our main contact in the company.

This final report is a summary for the project in whole and should reflect the work carried out during the work period.

2. INTRODUCTION

The goal of this project was to create an instant game for Facebook and mobile devices. Betware wanted us to create a game by using HTML5 which would be playable on modern mobile devices most importantly the iPhone and the iPad. Currently instant games are made with Flash, which is not supported on Apple mobile devices. They also wanted us to research a few things such as, the possibilities for Facebook integration with instant games and pushing notifications.

At first the goal was just to develop this instant game but after a discussion with Betware and because the number of participants in the project was greater than they had first estimated, it was decided that we would also develop a framework to facilitate the development of new games.

Most of the work took place in the Betware main offices at Holtasmári 1, Kópavogur. We decided to name our team as many high performance teams do and chose the name Betmen.

We made a vision statement which describes the main direction and goals for the whole project.

2.1. VISION STATEMENT

We are developing an Instant game that is written against Betware IGF (Instant Game Framework). Instant Game is a gambling game where the outcome is predetermined. All reusable code will go into a separate package (framework) which can be used to develop new games more easily.

The game is written in JavaScript and uses the benefits of HTML5. Unlike Flash games there is support for HTML5 and JavaScript on Apple's mobile devices (iPad, iPhone). The game will run in all major browsers and scale well between devices. The game can also make use of Facebook features.

3. ARTIFACTS

All artifacts of this final project are on the attached CD.

- **Source Code**
 - All code and resources for the project.
- **Facebook feature overview**
 - This document describes the possibilities for Facebook integration with instant games.
- **Final Report**
 - This document.
- **GCF Wiki**
 - Supporting document which describes all modules in the framework.
- **Operation Manual**
 - Document containing operating manual for the framework, how to create new games etc.
- **Product Backlog**
 - This document includes the product backlog and the sprint backlog along with the burndown charts for each individual sprint and the release burndown chart.
- **Project Structure**
 - Detailed work plan and structure for the project, how many sprints, how many hours of work etc.
- **Risk Analysis**
 - Document containing risk assessments for different parts of the project.
- **Software Design Document**
 - Document containing detailed explanation of the software design.
- **Testing Document**
 - Detailed document that describes all tests executed for the project.
- **Project Journal**
 - This document includes a log of the working hours for all team members.
- **User Manual**
 - This document describes the Fisherman game and its design.

4. METHODOLOGY

Right from the start we decided that we wanted to use Scrum to manage our project. All of us had some experience using Scrum and thought it was a good candidate for our project. We had never used Scrum in a big project before and therefore this was a good opportunity to use it. Betware also uses Scrum extensively so that made it very exciting.

Scrum is an agile methodology where the main focus is the outcome of the software development, not the process used in the project. We decided to have six two week sprints where at end of each sprint we would have working software.

Scrum has the following roles: Product Owner, he represents the voice of the customer which Hlynur Johnsen from Betware agreed to take on. The Scrum Master which is responsible for the product backlog and makes sure that the team gets privacy to do their work. We decided that Bergur Páll should be the Scrum Master. The Team includes all students participating in the project.

To break down the project we used a product backlog and a sprint backlog. The product backlog includes all stories for our project. Each story is given story points which represent the amount of work needed to finish the story. Our stories almost always indicated some feature in the project that could be tested with acceptance tests. Each story was broken down into tasks and put in the sprint backlog. This process is called sprint planning.

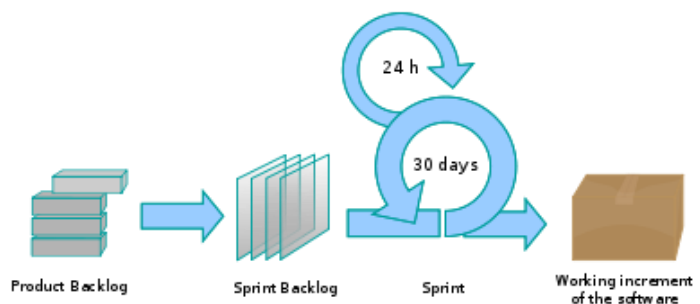


Figure 1: The scrum process (Anonymous, 2011)

Every day the team had a daily scrum meeting where the team told each other what tasks they had been working on, what tasks they are going to be working on next and if something was standing in their way. At the end of each sprint there was held a sprint demo meeting with Betware where the state of the project was shown. After the sprint demo we held a sprint retrospective where we talked about what went well or bad in the sprint and what to do better in the next sprint.

5. PROJECT ANALYSIS

5.1. WORK SCHEDULE

When we started working on this project we needed to be well organized so that the project wouldn't be affected by load in other courses. The team was participating in nearly all of the same courses and therefore it was easy for the team to work on the project at the same time. We estimated that each person would work 22 hours per week before the final exams and 40 hours per week after. We estimated that the whole project would be 1608 hours of work. More information about how we estimated the project in whole can be found in the *Project Structure* document.

At the end of the project we had worked on the project for 1671 hours where the average work time for each student was 418 hours. Overall we were satisfied with how much we could do in this time period. More information about actual work hours and what we were working on during each day can be found in the *Project Journal* document.

5.2. PROJECT PROGRESS

In this chapter we will explain the progress of the project as a whole. We managed our progress and project plan with Scrum and used an Excel spreadsheet for both the product backlog and the sprint backlog. Both backlogs can be found in the *Product Backlog* document.

5.2.1. SPRINT BURNDOWN CHARTS EXPLAINED

The red line stands for the estimated burndown in the sprint. It is based on the total hours at the end of the sprint. The blue line shows how the sprint actually progressed. The space between the lines at the top shows how many hours were added during the sprint.

The green bars show the actual hours per day, based on the project journal.

The x axis represents days in the sprint.

The left y axis is the hours for the red and blue burndown lines.

The right y axis is the hours for the green actual hour bars.

5.2.2. PLANNING PERIOD (JAN 12 – JAN 31)

The first days of the project went into preparations and us trying to get a clear view of the project scope. We agreed to use Scrum, laid out the project plan and important dates and milestones. We also wrote the projects vision, made a risk analysis and started work on the product backlog and user stories. At the end of the planning period we went to a summer house, where we worked on the project and had a good time. Generally our planning period went well and we finished all of our goals.

5.2.3. SPRINT 1 (FEB 1 – FEB 14)

Planned topics: Setup, design and research.

We had planned in cooperation with Betware to get access to their facilities on February 1st, which was however delayed until February 14th due to constructions. Because of these delays we had to move several stories and tasks to the next sprint, such as setting up the revision control.

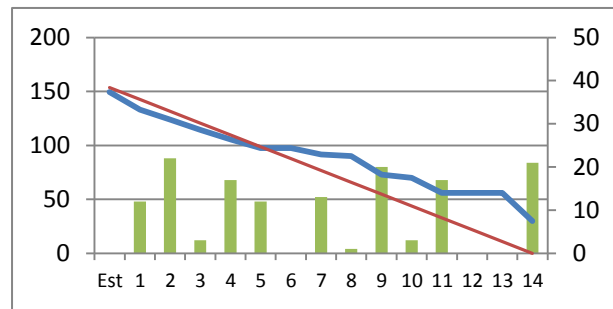


Figure 2: Sprint 1 Burndown

We established connection to the Betware systems, studied their Instant Game Framework SDK and made the Fisherman game concept.

5.2.4. SPRINT 2 (FEB 15 – FEB 28)

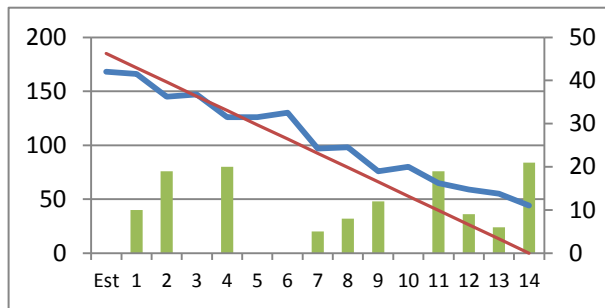


Figure 3: Sprint 2 Burndown

Planned topics: Framework and research.

The first work on the framework was to establish communication with the Betware IGF (Instant Game Framework) and have a HTML5 prototype with moving objects and basic scalability between devices. We also researched Responsive Web Design and possibilities for Facebook integration.

Communicating with the Betware IGF was easier than we planned because we had good documentation and code examples from Betware.

We had minor issues with how to implement event handling for user input in the framework and moved several tasks regarding events to sprint 3.

5.2.5. SPRINT 3 (MAR 1 – MAR 14)

Planned topics: Game, framework, Facebook and scalability.

We worked on the structure of the framework, events, audio, scalability, main loop and finished the config files for the IGF.

We got access to the deployment process at Betware in this sprint and therefore our continuous integration process was complete.

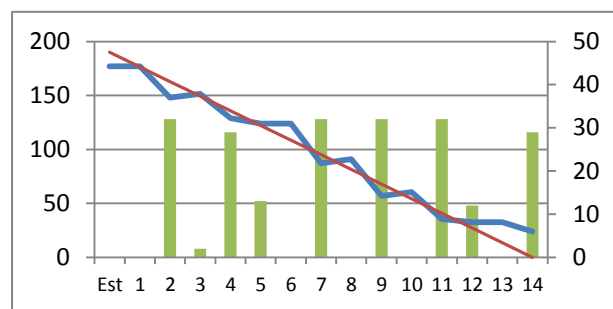


Figure 4: Sprint 3 Burndown

We started work on the Facebook authentication but we decided that it would not be a part of the framework so we postponed the story until sprint 4.

5.2.6. SPRINT 4 (MAR 14 – MAR 28)

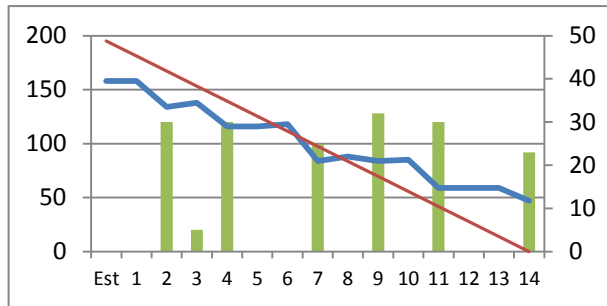


Figure 5: Sprint 4 Burndown

Planned topics: Fisherman, framework and Facebook.

In this sprint our focus was to start work on the Fisherman game. We also worked on audio and animations for the framework and had minor issues regarding those features.

Near the end of this sprint we went to the Westman Islands to consolidate the team and

boost the morale.

After this sprint was completed we had the basis for the framework and the Fisherman game. We could therefore go to our final exam period without much stress.

5.2.7. SPRINT 5 (APR 19 – MAY 2)

Planned topics: Fisherman, framework, Facebook and documentation.

This was the first sprint after the final exams and we could have full focus on the project. We started work every morning at 9 o'clock and finished at around 6 o'clock in the afternoon. We worked all Easter and only took one day off.

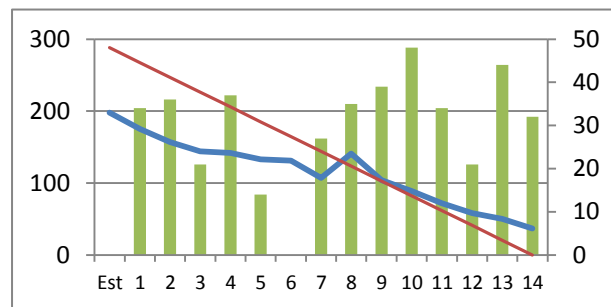


Figure 6: Sprint 5 Burndown

We were performing very well in this sprint and finished most planned features for both the framework and Fisherman. We also made a proof of concept for the framework by making another game (Breakout) and started work on some documents for the final status meeting.

5.2.8. SPRINT 6 (MAY 3 – MAY 16)

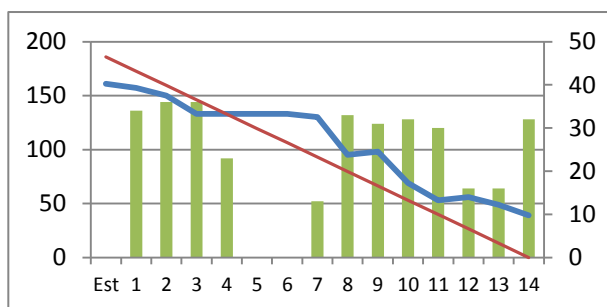


Figure 7: Sprint 6 Burndown

Planned topics: Code fixes, feature freeze, documentation and testing.

We finished work on Fisherman and the framework to make ready for delivery. We performed user tests with participants from outside the team. We went to the final status meeting which we prepared well for. The main focus in this sprint was however documentation for the final delivery of the project.

5.2.9. RELEASE BURNDOWN

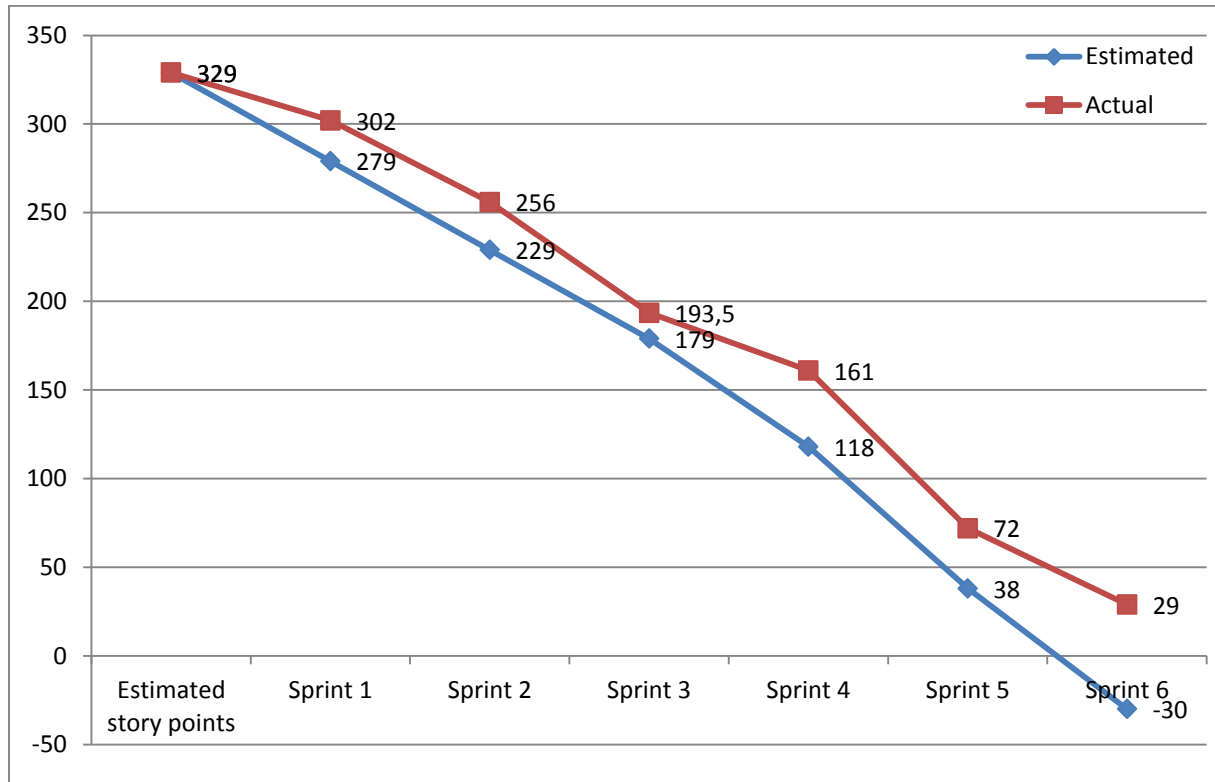


Figure 8: Release Burndown

In this chart you can see roughly how the project went. We were slightly behind schedule but all main features with the highest priority were implemented. During the project we added and removed irrelevant stories. The added stories can be seen in this chart as they appear below the zero mark. That sums up to 30 story points during the whole project. Those added stories were all completed.

29 planned story points remain in the backlog as they could not be finished on time. Those stories are all in priority C (level 4).

5.3. RISK ANALYSIS

In the beginning of the project we performed a risk analysis to find out what could be most threatening factors for the project and how we could prevent them for happening.

We found many possible risks and estimated when and how we could resolve the risk and marked it resolved on the day it was resolved.

All risks were assigned with a risk factor and a probability. The risk factor is a measurement of how harmful the risk would be to the project if it happened, the risk factor ranged from 1 to 5. The probability ranged from 0% - 100%.

We calculated the risk factor by multiplying the risk factor and the probability. In the total risk we get a number from 0 – 5 where 5 is the most threatening risk.

When the project was finished we found out that some of our risks were not estimated correctly. Some risks were estimated to high, like little experience in JavaScript because we all have so good foundation in programming in general. And other risks were estimated to low like application lagging in Apple mobile devices which caused us more problem than we anticipated.

In the end of the project we had eliminated all the risks, the last one being marked resolved May 16th on the day of hand in.

The complete risk analysis can be found in the *Risk Analysis* document.

6. DESIGN AND DEVELOPMENT

Overall design and development process went well in our project because we were very organized from the beginning and had good experience from school. We were very inspired by the use of software patterns, which we tried to implement in our project where it was suited.

To keep our documents organized and available at all times we used Dropbox. Dropbox is a free file sharing system that has been of good use. The problem with the system is that only one person can work on a document at any time. We resolved that by working in pairs in separate documents.

Right from the beginning we decided to use subversion for our code so that we could branch, merge and be working in the same code at the same time. Our project was branched five times on different stages of the project. We did not have to set up our own subversion because Betware provided us with access to theirs.

We used a build script to wrap our project. Betware is using an application called Bamboo from Atlassian which is a continuous integration build server that they gave us access to and helped us setup. Each time we committed code to the subversion Bamboo built that version to be readily deployed on our test machine.

To deploy our build we only had to connect to our test machine and execute one line. That saved us a lot of time in the development process. Otherwise we either would have had to set up our own build and deployment process which would have taken a lot of time, or doing things sloppy by copying the files to the server.

6.1. FISHERMAN

It was up to us to design the game, Betware wanted new and fresh ideas for the game we were going to develop. At first we were struggling with the decision about what game we were going to develop, it was difficult because of the constraint the predetermined design of the IGF. However we finally decided to make a fishing game which we called Fisherman. In the game you have five throws to try to catch three fishes of the same kind. If you catch three fishes of the same kind you get a price which depends on how much you bet and what kind of fish you caught. Detailed design description of the game and the game rules can be found in the *User Manual* document.

6.1.1. GRAPHICS

It was decided early on that Betware would provide us with graphics for the game. When the game design was finished on our part we made some sketches for the game.

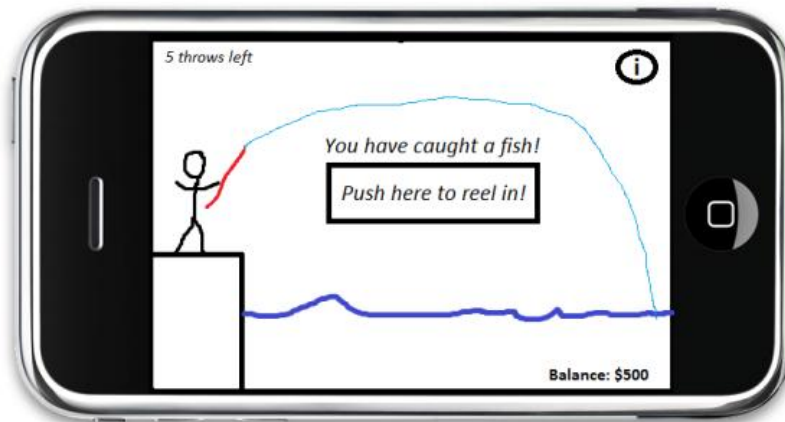


Figure 9: Early Fisherman sketch

We met with their graphic designer Þorsteinn Hreggviðsson in late March and informed him about our game design and showed him our sketches. He suggested that we would make a mood board so he could get a feeling about what kind of theme the game should have.



Figure 10: Mood board for Fisherman

Þorsteinn then made the basic graphics for the game based on our mood board and sketches. There were no animations apart of the graphics he provided us with. We needed animations for a few things like the fisherman, for when the fishes come out of the water etc. In the end we made the animations for the game ourselves, it took us a lot of time because none of us has much experience with graphics software like Photoshop. But in the end we were satisfied with the result.

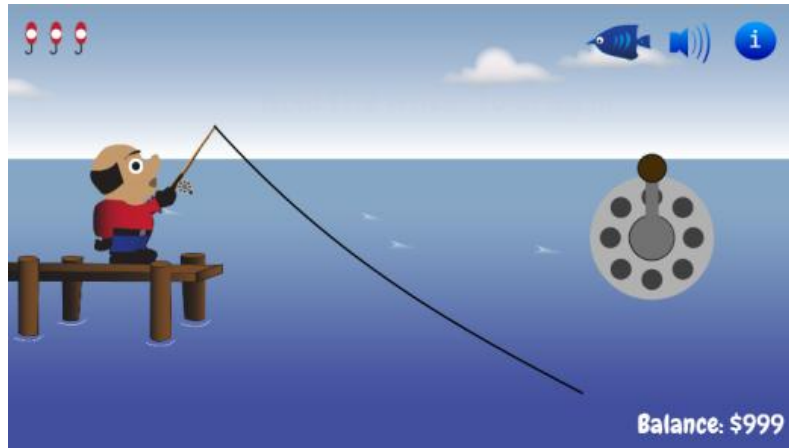


Figure 11: Final Fisherman design

6.1.2. SKINNABILITY

With our framework it is very easy to change the skin of the game. All that the developer has to do is to create another resource folder with all the same resources that he used in the original skin. Then to use the new skin, he either has to change the value in the xml file or send the new skin name into the code as a parameter. This is all very simple as we intended it to be. More information can be found in the *Operating Manual* document.

For our game Fisherman we did an extra Christmas theme skin which took little time.



Figure 12: Christmas Skin

6.1.3. FACEBOOK

A part of this project was to find out how we could use Facebook to enrich the user experience in our game. One of the first things we did was to research those possibilities, the results from that research can be found in the *Facebook feature overview* document.

For the Fisherman game we decided to make it possible for players to sign in to Facebook to get information about the player and make it possible for him to post results from the game to his wall. We also discussed with Betware to implement a compare to friends feature but due to the predetermined nature of the IGF there are no scores involved, the player either wins or loses. It was not logical to implement that feature, it would also mean a lot more work for us because we would have to get access to a database and handle the communication between the game and the database.

There are more possibilities we could have implemented if the game would be hosted/accessible on Facebook such as request and bookmarks. But because Facebook does not allow gambling application on their site these features could not added.

6.2. GAME CLIENT FRAMEWORK (GCF)

Our design goal for the framework was to have it well organized and easy to use. We split the functionality of the framework into modules which all have their specific purpose.

- **Game Client Framework**
 - A base module that includes the other modules and loads them into memory.
- **Audio Module**
 - The Audio Module handles audio and volume.
- **Communication Module**
 - The Communication Module handles the communication between the client and the Betware Instant Game Framework (IGF).
- **Event Module**
 - The Event Module handles events, such as mouse, keyboard and touch events. The module also provides a few optional functions which can handle event for you.
- **Game Module**
 - The Game Module handles the canvas creation, mapping the content between different canvas sizes, orientation change, the main loop, amongst other things.
- **Platform Module**
 - The Platform Module is used to detect the platform the player is currently using.
- **Resource Module**
 - The Resource Module handles the loading of resources.
- **Utility Module**
 - The Utility Module is mainly a class factory for draw able objects such as scenes and different kinds of sprites.

More information about the GCF can be found in the *GCF Wiki* and the *Software Design Document*.

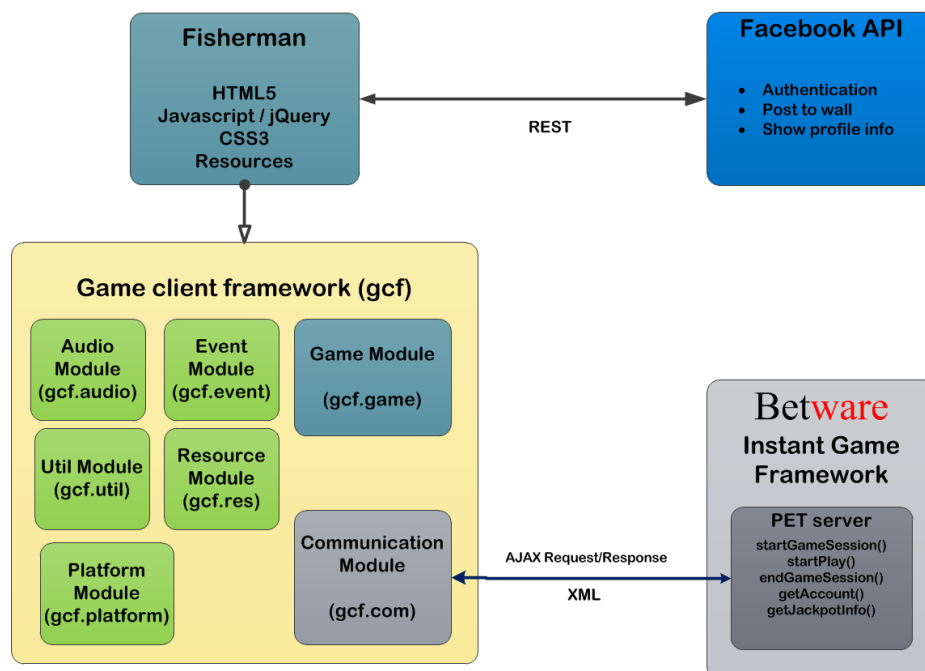


Figure 13: Overview of the GCF framework and Fisherman

6.2.1. BREAKOUT

To show how easy and simple our framework is to use, we made another game in portrait as a proof of concept for the framework. The game we decided to develop is called Breakout and is a totally different from Fisherman and doesn't connect to Betware IGF. Breakout took less than one day for one of us to develop and is 260 lines of code. More information on how to make a new game can be found in the *Operating Manual* document.

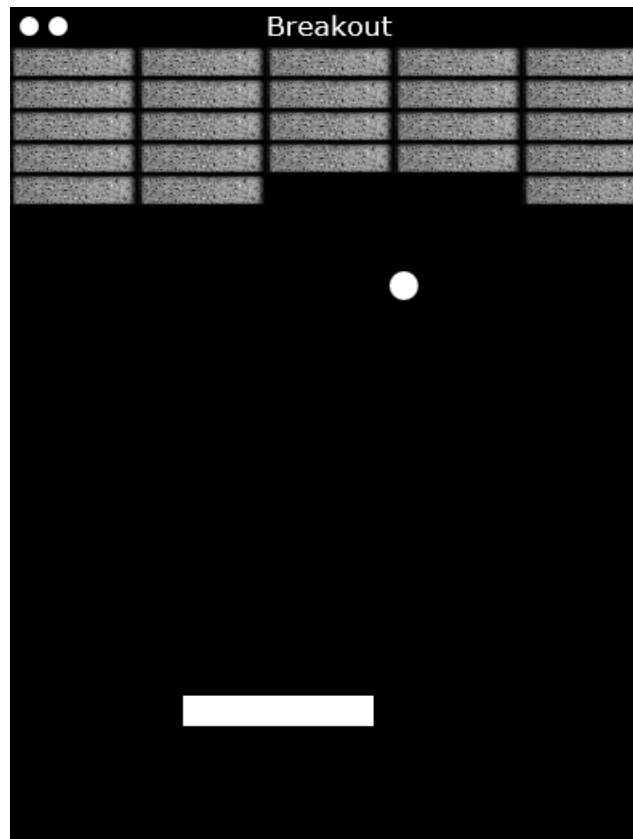


Figure 14: Screenshot of Breakout

6.3. TESTING

While developing the game client framework and the fisherman game we performed three kinds of tests, unit tests, user tests and acceptance test.

6.3.1. UNIT TESTS

We used unit testing to test the modules of the GCF. Fisherman was not tested specifically because it relies heavily on the game client framework which is thoroughly tested. All other code in fisherman is game logic which is hard to test.

We used Qunit to unit test the code where the unit tests were executed manually to a specific website which was included in the build.

6.3.2. USER TESTS

We performed user tests for the fisherman game to ensure that the game was bug free, performing well and was easy to learn. Each one of us got one person to test the game by going through all the game in steps and the user was supposed to do it without help.

We tested the game on desktop, iPad and iPhone to ensure that the game works well on all devices. We then collected all the data from the test and fixed all the bugs and suggestions that we were given.

6.3.3. ACCEPTANCE TESTS

When a new feature was added in the game and committed to subversion we tested it on all the devices that the game was designed for such as browser, iPad and iPhone. In some user stories we had a „verify that” clause, and that acted as a guideline for the acceptance test.

6.3.4. RESULTS

In total we executed 126 unit tests in 37 categories. We are currently covering 39.5% of our code with unit tests. We would be covering more but it is difficult to unit test many of our functions, because they involve drawing logic or and need to have canvas on the page.

From our user test we found 14 bugs/comments which we have all fixed or handled.

More information about testing can be found in the *Testing* document.

7. SUMMARY

Overall this final project went very well, due to the fact that we were very organized from the beginning, everything went smoothly and we finished almost everything we set out to finish, as you can see in the *Project Structure* document and the *Product Backlog*.

From the beginning we knew that there were a few risks associated with this project, for example using new technologies such as HTML5, to scale the game between different devices and amongst things. We also knew that none of us had any programming experience in JavaScript. Therefore a large portion of our time went into researching these technologies and finding the best way to proceed.

There are a few things that we are very satisfied with and proud of, like how successfully we implemented the framework and how easy it is to program new games, as you can see we developed two games, first and foremost Fisherman which our end result is very well aligned with what we set out to do and secondly Breakout which can be thought of as a proof of concept for the framework. We are also very proud of how the framework handles scaling, resource management, scene/sprite setup and skinnability.

There were also some things that became more difficult than we had anticipated. Things like support for audio on Apple mobile devices was terrible, also the performance on these devices was not satisfactory until we had tweaked it.

Our communication with the Betware representatives associated with our project was exemplary, they influenced our decision-making during sprint meetings and sprint reviews. They also provided access to their continuous integration setup which we used throughout the project.

If we had more time to work on this project we would add more features to the framework such as, push notifications and collision detection. We would also want to develop a new game which would be better suited for implementing Facebook features.

8. CONCLUSION

In the end we are very happy with the result of our project and we feel that we earned valuable experience while studying a lot of new topics, for example we had never used Scrum in big project before, we learned a lot about game and framework development and finally we learned a lot about HTML5 especially drawing with the Canvas. All these experiences will come to good use in the future. We hope that Betware is happy with the results and can use some of our ideas in their work in the future.

In the end we want to thank all of those who helped us during this project. We want to thank our instructor Hannes Pétursson for all his help and tips he gave us during our meetings. We also want to thank Hlynur Johnsen, Sigtryggur Arnar Árnason, Jónas Þór Jónasson, Þorsteinn Hreggviðsson and the operation department at Betware for all their help and patience. We also want to thank Betware for the opportunity to get the experience of real software development process. Finally we want to thank Hlynur Sigurbórsson examiner and Marta Lárusdóttir supervisor of the final projects for good advices from our status meetings.

8.1. REVIEW FROM BETWARE

Betmen, a team from Reykjavík University worked on a multi-platform game project for Betware for a period of 4 months from January 2011 as part of their final project. The game was built using a Betware gaming API and was optimized for running on iPhone and iPad as well as a PC browser. The team was responsible for the design and implementation of the game, as well as a framework for building JavaScript games that connect with Betware gaming API and a JavaScript game development utility package.

The team worked independently on the project within Betware's headquarters. The work was extremely well organized which made tracking the project very accessible to the Betware support staff. Despite limited resources, the team solved all the problems they were faced with and delivered a very nice looking and entertaining, near-production quality game, built on well designed, well implemented and well documented technology.

Overall, Betware is extremely pleased with the progress the team made during the execution of the project as well as the final result. Betmen are clearly a group of very competent software engineers and Betware is happy to have had the privilege of accommodating the team during their final project at Reykjavík University.

On behalf of Betware

Hlynur Johnsen
Product Manager

Sigtryggur Arnar Árnason
Development Team Lead

9. BIBLIOGRAPHY

Wikipedia contributors. (2011, 05 10). *Wikipedia*. Retrieved 05 10, 2011, from Scrum:
[http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))

Atlassian. (n.d.). Retrieved 05 13, 2011, from Bamboo:
<http://www.atlassian.com/software/bamboo/tour/>